Using Semantic Role Labeling to Extract Events from Wikipedia

Peter Exner and Pierre Nugues

Department of Computer science
Lund University
peter.exner@cs.lth.se
pierre.nugues@cs.lth.se

Abstract. Although event models and corresponding RDF vocabularies are becoming available, the collection of events still requires an initial manual encoding to produce the data. In this paper, we describe a system based on semantic parsing (SRL) to collect automatically events from text and convert them into the LODE model. Furthermore, the system automatically links extracted event properties to the external resources DBpedia and GeoNames. We applied our system to 10% of the English Wikipedia and we evaluated its performance. We managed to extract 27,500 high-confidence event instances. Although SRL is not an error-free technique, we show that it is an effective tool, as the definition of the arguments (or roles) used in our analysis and the event properties are, most of the time, nearly identical. We evaluated the results on a randomly selected sample of 100 events and we report F-measures of up to 73. The extracted events are available online from a SPARQL endpoint[1].

1 Introduction

Event models, such as EVENT[2] [1], LODE[3] [2], and SEM[4] [3], share common features to represent the agents, time, and place involved in an event. Such models are interesting because they attempt to reconcile disparate theories and standardize their representations using RDF vocabularies; for a review and a discussion, see [3]. Ideally, they should enable a variety of providers to publish any kind of events in distributed repositories, where clients would gain a uniform access to data. Applications could then embed more easily event-related information and processing. However, actual mentions of events in source materials, such as history books, newspapers, encyclopedia, etc., rarely comply with such representations. Before we can get access to wide-coverage and standardized event repositories, we need to find ways to automate their collection – i.e. their detection and extraction – as well as the identification of their properties from these source materials.

Event calendars available from websites such as Last.fm, upcoming.yahoo.com, and eventful.com, are accessible through application programming interfaces (API). These

[1] http://semantica.cs.lth.se/
[2] http://motools.sourceforge.net/event/event.html
[3] http://linkedevents.org/ontology/
[4] http://semanticweb.cs.vu.nl/2009/11/sem/

calendars provide events in a structured format in which the majority of the event properties, such as time and place, has already been extracted. Transforming these calendar data to a given event model can be done through the mapping of their format to the properties of the selected model as in [4]. Extracting events from natural language, as found on blogs and ordinary web pages, poses a greater challenge since the events are inherently unstructured.

In this paper, we introduce a system to extract events automatically from natural language using semantic parsing. We built a processing pipeline that takes raw text as input and extracts predicate–argument structures from the sentences. We used a semantic role labeler (SRL) to identify the predicates together with their core arguments or roles, such as the agent or the theme, in the sentences. The predicate arguments also include modifiers, such as temporal, locational, and manner adjuncts.

Semantic role labeling [5] is a generic technique to parse predicate–argument structures, where most of the semantic role labelers for English use statistical models trained on either Framenet [6] or Propbank [7]. Although they can reach acceptable levels of performance in terms accuracy [8, 9], semantic role labelers are often too slow to be applied to large corpora as is, or lack specificity to be used in dedicated information extraction tasks. In the context of SRL, the extracted predicate–argument structures are often called *propositions*. We will use this term in the rest of the paper.

To gather a significant set of events, we used the English Wikipedia[5] as the source material. In addition to being sizable and easy to access, Wikipedia has a large coverage of historical and cultural events that, we believe, cannot be matched by other corpora. To cope with the size of this corpus, we extended the core SRL system with a database to store the propositions and backlinks to their location in the source text. Conceptually, the extraction of events comprises four main stages:

1. Semantic parsing of Wikipedia (SRL);
2. Event selection: argument identification and property extraction;
3. Disambiguation and linking of the time and location phrases to external resources;
4. Mapping of the predicate–argument structures onto an event model.

We evaluated the performance of our system to identify and extract events. We show that SRL is an effective tool, as the definition of the Propbank arguments (or roles) used in our analysis and the event properties as described in the event models are, most of the time, nearly identical.

2 System Architecture

The architecture of our event extraction system is a pipeline of components. It consists of four main modules (Figure 1):

1. The parsing module, Athena, is a framework for large-scale parsing of text written in natural language;
2. The argument identification module that associates the predicate–argument structures extracted by the first module and relates them to a restricted set of VerbNet roles;

[5] `http://www.wikipedia.org/`

3. The property extraction and linking module that associates agent, time, and location phrases to GeoNames and DBpedia entries;
4. The conversion module that maps the structures to the LODE event model.

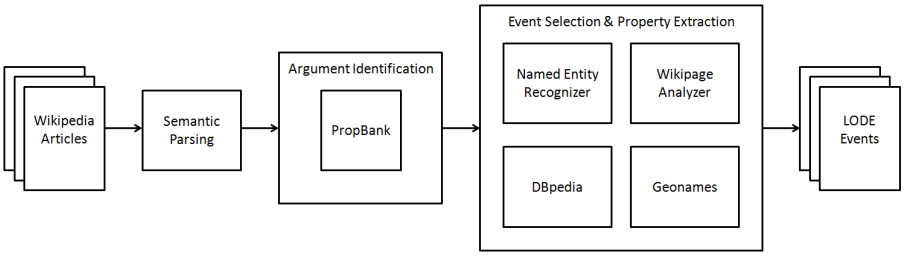

Fig. 1. Overview of the event extraction pipeline.

3 Semantic Representation of Sentences

Semantic Roles and Event Models. There are many linguistic theories on the semantic representation of sentences. Frame semantics [10] is one of the most productive that assumes that the meaning of a sentence is represented by a set of predicates and arguments. Framenet [6] and Propbank [7] are two projects that applied this theory to annotate corpora, respectively the British National Corpus and the Wall Street Journal with their predicate-argument structures. Predicates can have different senses, where each sense is associated with a specific set of arguments.

The argument annotation goes beyond the traditional subject and object and includes modifiers of the predicate, such as the temporal, locational, and manner adjuncts. These modifiers are crucial in the extraction of events since all the event models contain properties to hold the time and the place.

Figure 2 shows the predicate and the arguments contained in the sentence *In 1953, John Desmond opened the first architectural firm in Hammond.* annotated using the Propbank style. The predicate *open.01* uses the suffix 01 to denote its sense that corresponds to *open*. This differs from *open.02*, which means to *begin*. The A0 argument, *John Desmond* and the A1 argument, *the first architectural firm*, have the meanings *opener* and *thing opening* respectively for this predicate sense. The phrases *in Hammond* and *In 1953* correspond to locational and temporal modifiers, AM-LOC and AM-TMP, respectively. An ideal mapping would assign the core arguments A0 and A1 as well as the modifiers AM-LOC and AM-TMP to the agent, time, and place properties of an event model. In addition, proper nouns can be extracted, disambiguated, and linked to external resources.

The LODE Event Model. We chose the LODE event model to represent our extracted events because LODE is independent of the event domain, does not force aspect or agentivity, and makes a distinction between a named place and a geospatial space. We

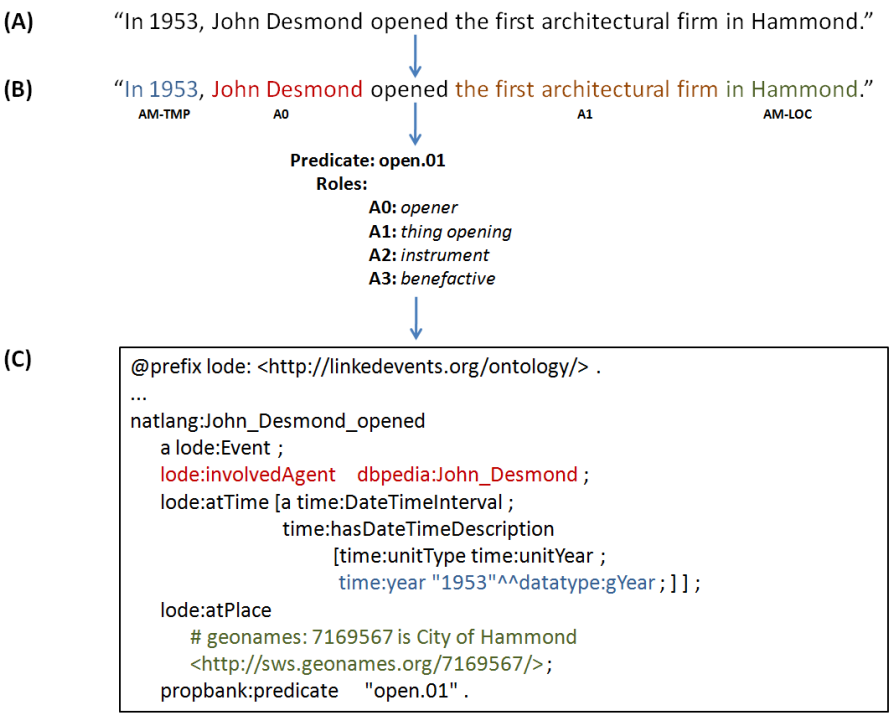

Fig. 2. (A) The starting sentence. (B) The sentence after parsing with SRL. (C) An ideal conversion to the LODE event model.

believe these features necessary for representing the wide diversity of the events found in Wikipedia. Compared to SEM, it is a minimal model that fits generally well with the conceptual nature of information contained in natural language sentences. Figure 2 shows an example of an ideal transformation of a predicate–argument structure to the LODE event model.

Semantic Parsing. As core parser, our system uses a high-performance multilingual semantic role labeler that obtained top scores in the CONLL-2009 shared task [9].

Even if SRL has made significant progress during the last ten years, it is still prone to errors especially with phrases involving proper nouns and adjuncts. This makes it more difficult to apply it to event extraction as events contain inherently more proper nouns than other sentences, both in the description of the event and the place. In addition, proper nouns like *Research in Motion* may contain words where the parser can misclassify a word as a verb, here *Research*, and then lead to a wrong predicate extraction. Furthermore, time expressions and time intervals are still not perfectly identified. Although predicate–argument extractions are not done with 100% accuracy, we still believe that SRL can be a very useful tool for event extraction and we also propose workarounds to increase the quality of the extractions.

4 Athena

Athena is a parsing framework intended to cope with large-scale multilingual information extraction. It consists of several components that fill a specific task in reading the Wikipedia text including both the English and Chinese versions, extracting, analyzing, and transforming knowledge. By using trained parsing models, the framework can be adapted to new languages without the need of reworking the extraction algorithms or patterns.

Athena reads articles from a Wikipedia database, filters, parses, and then stores the data in a semantically annotated structure. The task of parsing the entire database is parallelized using scripts, which subdivide a range of articles and launch parsing jobs applied to smaller ranges. Athena builds the proposition database by gathering the multiple small databases created during parsing and assembles them into one large database. With the use of a statistics module, the proposition database can be queried to provide statistics such as the number of and redundancy of propositions.

In our experiments, we used a subset of 10% of the English edition of Wikipedia consisting of 378,453 articles. We extracted all the sentences of all the articles and we parsed them. It resulted into 13,428,114 sentences and 53,694,899 propositions. We believe this size to be large enough to provide a significant number of propositions and events and at the same time enable us to carry out a sequence of try-and-fail experiments with an acceptable cycle time.

Mapping Predicates onto Events. Although predicates and events, such as in Probbank and LODE, have a similar structure, they are not identical. A major difference is that a set of arguments in Propbank is specific to one predicate sense, for instance the arguments of *open.01* are A0, *opener*, A1: *thing opening*, A2: *instrument*, and A3: *benefactive*, while LODE has only two universal properties, **involved** and **involvedAgent**, that correspond to these Propbank's core arguments. To cope with Propbank's diversity, a converter is necessary to map the predicate–argument structures onto the selected event model. [11] is an example of this that uses hand-generated rules or rules induced from manually-filled event templettes.

Instead of using rules that in any case would require significant manual work, we took advantage of the links between Propbank and VerbNet and we implemented a mapping module based on it. VerbNet [12] is a lexicon that builds on Levin's classification of English verbs [13]. Verb classes are described using a limited set of 23 roles used across all the lexicon and where each predicate role is constrained using selectional restrictions such as animate, comestible, etc. Although not complete, 11,500 arguments in Propbank have a correspondence with VerbNet thematic roles, making the conversion possible.

5 Selecting Event Propositions

We built our event set from the complete proposition output produced from Wikipedia. We considered that a proposition could fit an event if it contained a date, a place, and an agent. For a discussion on the aspects of event classification, see [14]. We used the links

associating the Propbank arguments to the VerbNet thematic roles and we extracted the propositions whose arguments matched a time, a place, and agents in the VerbNet structure. We used the following rules:

- We identified an agent from a Propbank argument when it could be associated with one the following VerbNet thematic roles: *Actor*, *Agent*, *Beneficiary*, *Experiencer*, *Recipient*, and *Theme*. If no such roles were found, we selected the *A0* argument as default.
- Similarly, we identified the places using the *Location* and *Source* VerbNet thematic roles. We also included the *AM-LOC* modifier.
- We could not find arguments in PropBank linked to the *Time* VerbNet thematic role. We therefore selected the arguments containing dates and times using the *AM-TMP* modifier.

These events were further filtered by selecting propositions having at least one extracted time, place, and agent property. Using a quick manual examination, we could observe that this very simple filtering enabled us to discard a large set of less reliable propositions.

6 Converting Propositions to Event Models

Following the argument identification, we extracted entities corresponding to the LODE ontology properties using regular expressions, a local subset of the DBpedia database [15], and the GeoNames web service[6].

Aspectual Verbs. We grouped pairs of predicates that begins with an aspectual verb, such as in *began working* or *stopped singing*. This grouping was performed when the second predicate together with all of its arguments formed a subset of the arguments of the first predicate. Figure 3 shows an example of it, where the arguments of the predicate *work.01* form a subset of the arguments of the predicate *begin.01*. Thus, the two predicates are grouped to form the event, *began working*.

	In	the	late	1990s	NASA	and	Google	began	working	on	a	new	network	protocol	.
begin.01	AM-LOC				A0				A1						
work.01					A0					A1					
protocol.01												AM-TMP	A1		

Fig. 3. Parsing output showing an example of a sentence, where we group two predicates: *In the late 1990s NASA and Google began working on a new network protocol*. The semantic parser is accessible from `http://barbar.cs.lth.se:8081/`.

Single predicates and predicate groups are assigned to the *propbank* RDF property.

[6] `http://www.geonames.org/`

Converting Involved Agents. When possible, we linked the LODE arguments to DBpedia entries. This enabled us to integrate the data we produce with other types of structured information extracted from Wikipedia and from other sources. Eventually, this should improve interoperability of data sources and make it easier to build comprehensive applications.

To detect the entries, we applied a named entity tagger[7] [16] to the arguments extracted from the VerbNet thematic roles. We then selected entities representing organizations and persons as agent candidates. We used a subset of the DBpedia database containing infobox types, Wikipedia redirects, and Wikipedia page links to carry out the final name disambiguation.

Candidates are disambiguated and linked to their corresponding DBpedia entry by one of the following rules in this order:

1. When an infobox type matches the candidate phrase, we use this type. For instance the phrase *United Nations* is resolved directly to the DBpedia resource `<http://dbpedia.org/resource/United_Nations>`;
2. When a redirection is found for the candidate phrase, we use this redirection. As an example, the phrase *United States Supreme Court* is resolved to the DBpedia resource `<http://dbpedia.org/resource/Supreme_Court_of_the_United_States>` by using DBpedia page redirects;
3. When outgoing DBpedia resources from the originating Wikipedia article contain the candidate phrase, we use the most frequent resource. For example, if we wish to resolve the word *Loren* in the Wikipedia article `http://en.wikipedia.org/wiki/Carlo_Ponti` to a DBpedia resource, we start from the originating article and consider only outgoing DBpedia resources that contain the sought phrase. We find that the DBpedia resource, `<http://dbpedia.org/resource/Sophia_Loren>`, is mentioned three times in the article and we select this as the resolved resource for the word *Loren*;
4. When labels of outgoing Wikipedia links from the originating Wikipedia article also contain the candidate phrase, the corresponding targets are selected and resolved using the rules above. In this case, the outgoing DBpedia resources do not contain the candidate phrase and the labels of the outgoing Wikipedia links are searched instead. Using this technique, *Edith Somerville* is resolved to the DBpedia resource `<http://dbpedia.org/resource/Edith_Anna_Somerville>` in the Wikipedia article `http://en.wikipedia.org/wiki/Violet_Florence_Martin`;
5. When the title of the originating Wikipedia article contains the candidate phrase, it is selected and resolved using the rules above. For instance, *Weis* is resolved to the DBpedia resource `<http://dbpedia.org/resource/Weis_Markets>` in the Wikipedia article `http://en.wikipedia.org/wiki/Weis_Markets`.

If a DBpedia entry is found, it is assigned the *involvedAgent* property in the LODE model.

[7] `http://nlp.stanford.edu/ner/index.shtml`

Converting Place and Space. Similarly to the extraction of the involved agents, we identify arguments corresponding to the VerbNet thematic roles associated with locations. Following named entity extraction, entities representing locations are queried using the GeoNames web service. The first matching result is selected and the GeoNames identifier is assigned to the *atPlace* property. Entities representing organizations are identified and linked to DBpedia entries by using the methods 1 to 3 described in Section *Converting Involved Agents*.

Converting Time. The conversion to the LODE *atTime* property is carried out in 3 steps: We first identify the arguments containing date and time phrases; We then extract the time entities from the arguments using the named entity tagger; And we finally convert the time entities to the OWL DateTimeInterval format using of common date format patterns. In addition, we discard time phrases without an anchoring date expression, such as *Three days ago*. Our extraction module identifies the first occurring date expression and assigns it to the *atTime* event property.

Storing Events. The extracted events are saved to files in the Notation 3 format. The file names contain the Wikipedia article title, followed by the absolute line number of the sentence from which the event was extracted. This structure enables the backlinking from the event to the originating source material.

7 Experimental Results

In our evaluation, we sought to answer the questions: How much of the information in a sentence can be extracted and moved into an event model? And, which properties are the most difficult to extract? Since we did not have the precision or the recall of events in the source text, we omitted the evaluation of event identification and instead we focused on calculating the precision and recall of the identified and extracted events. We approached this task by computing the recall and precision of the individual properties of our extracted events and counting the error sources.

In total, we extracted 27,594 events from our subset of 378,453 English Wikipedia articles. We created our data set by randomly selecting a sample of 100 events from our extracted events. In order to calculate precision and recall, we calculated the number of retrieved atTime, atPlace, involvedAgent, and predicate properties in each sampled event. We examined the sentence corresponding to the sampled event to find the number of relevant properties.

We used two metrics to assess the properties using a strict and a relaxed criterion. We marked the atTime property as strictly correct if all the date components were extracted, and as relaxed correct, if the most significant date component was extracted. We marked the atPlace property as correct if the extracted reference to GeoNames or DBpedia was resolved to a correct entry. We made no distinction between strictly correct and relaxed correct for atPlace. Similarly, we marked InvolvedAgent as correct, if the property resolved to a correct DBpedia entry. Finally, we marked the predicate

property as strictly correct if both the corresponding verb and sense had the correct semantics and as relaxed correct regardless of the predicate sense. During evaluation, we also counted the properties causing the errors.

Based on these evaluations, we calculated the precision, recall, and the F1 score for our sample data set (Table 1, left). Table 1, right, shows the relative percentage of error sources categorized by extracted properties.

	Precision	Recall	F1
Strict	70.8	71.6	71.2
Relaxed	72.8	73.6	73.2

Error sources	
Agent	40.9%
Place	36.9%
Time	11.4%
Predicate	10.7%

Table 1. Left table: The precision, recall, and F1 score for the sampled events. **Right table**: Sources of errors.

From Table 1, we can observe that the largest percentages of error come from the agents and places. The reasons for these extraction failures can be attributed to the following causes:

- The arguments containing the agents were not found by the semantic parser.
- Ambiguity of the extracted proper nouns.
- Unresolved pronouns.
- Lack of DBpedia entry corresponding to the agent.

We believe that in many cases the ambiguity of the agents can be resolved by using a larger subset of DBpedia databases and thereby classifying the type of the agents. Together with a more explorative selection of arguments, we believe this may lead to a larger amount of correctly extracted agents.

Since we only extracted the first detected date, this caused the majority of failures in date extraction. We believe that date extraction can be improved using more extraction patterns.

8 Conclusions

In this paper, we investigated semantic parsing to extract events from text. We implemented a processing pipeline consisting of a high-performance semantic role labeler to extract predicate–argument structures and a converter using VerbNet thematic roles to produce events in the LODE RDF format. Using 10% of the English Wikipedia and simple filtering rules, we managed to sift more than 27,500 high-confidence instances. We evaluated the results on a randomly selected sample of 100 events and we report F-measures ranging from 71.2 to 73.2.

Misidentified agents are a frequent source of error. We believe such errors can be significantly reduced by improving the detection of proper nouns. This could be done by applying a preprocessing step to detect the named entities or using databases of proper

nouns and retraining the semantic parser on it. We could also improve the detection using a coreference solver that would tie pronouns such as *she*, *he*, or *it* to person or organization names. In the future, we also plan to parse the complete Wikipedia corpus in English and other languages.

An archive of extracted events is available for download as well as accessible from a SPARQL endpoint[8].

Acknowledgments. This research was supported by Vetenskapsrådet, the Swedish research council, under grant 621-2010-4800 and has received funding from the European Union's seventh framework program (FP7/2007-2013) under grant agreement 230902.

References

1. Raimond, Y., Abdallah, S., Sandler, M., Giasson, F.: The music ontology. In: Proceedings of the International Conference on Music Information Retrieval, Vienna (2007)
2. Shaw, R.B.: Events and Periods as Concepts for Organizing Historical Knowledge. PhD thesis, University of California, Berkeley (2010)
3. van Hage, W.R., Malaisé, V., Segers, R., Hollink, L., Schreiber, G.: Design and use of the simple event model (SEM). Journal of Web Semantics (2011) In press
4. Liu, X., Troncy, R., Huet, B.: Finding media illustrating events. In: Proceedings of the 1st ACM International Conference on Multimedia Retrieval. ICMR '11 (2011) 58:1–58:8
5. Gildea, D., Jurafsky, D.: Automatic labeling of semantic roles. Computational Linguistics **28** (2002) 245–288
6. Ruppenhofer, J., Ellsworth, M., Petruck, M.R.L., Johnson, C.R., Scheffczyk, J.: Framenet ii: Extended theory and practice. `http://framenet.icsi.berkeley.edu/` (2010)
7. Palmer, M., Gildea, D., Kingsbury, P.: The Proposition Bank: an annotated corpus of semantic roles. Computational Linguistics **31** (2005) 71–105
8. Johansson, R., Nugues, P.: Dependency-based syntactic–semantic analysis with PropBank and NomBank. In: Proceedings of CoNLL-2008, Manchester (2008) 183–187
9. Björkelund, A., Hafdell, L., Nugues, P.: Multilingual semantic role labeling. In: Proceedings of CoNLL-2009, Boulder (2009) 43–48
10. Fillmore, C.J.: Frame semantics and the nature of language. Annals of the New York Academy of Sciences: Conference on the Origin and Development of Language and Speech **280** (1976) 20–32
11. Surdeanu, M., Harabagiu, S., Williams, J., Aarseth, P.: Using predicate-argument structures for information extraction. In: Proc. of the 41st Annual Meeting of the ACL. (2003) 8–15
12. Kipper-Schuler, K.: VerbNet: A broad-coverage, comprehensive verb lexicon. PhD thesis, University of Pennsylvania, Philadelphia (2005)
13. Levin, B.: English verb classes and alternations: A preliminary investigation. University of Chicago Press, Chicago (1993)
14. Shaw, R., Troncy, R., Hardman, L.: LODE: Linking open descriptions of events. In: 4th Asian Semantic Web Conference. (2009) 153–167
15. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: DBpedia—a crystallization point for the web of data. Journal of Web Semantics (2009) 154–165
16. Finkel, J.R., Grenager, T., Manning, C.: Incorporating non-local information into information extraction systems by Gibbs sampling. In: Proc. of the 43rd Annual Meeting of the ACL. (2005) 363–370

[8] `http://semantica.cs.lth.se/`